190 (approximate search) Given a nonempty sorted list of numbers and a number, write a program to determine the index of an item that is closest in value to the given number.

After trying the question, scroll down to the solution.

§ Linear search is easiest, but the list is sorted, so let's make it a binary search. Let the list be L , let the number be x , and let the natural variable whose final value will be the index of a closest value be h . We also need natural variables i and j . The problem specification R , and specification Q , are defined as

$$R \;=\; abs\ (x - L\ h') \;=\; (\Downarrow k\colon 0,..\#L\cdot\ abs\ (x - L\ k))$$

$$Q \;=\; abs\ (x - L\ h') \;=\; (\Downarrow k\colon h,..j\cdot\ abs\ (x - L\ k))$$

Here are the refinements.

$$R \;\Leftarrow\; h{:=}\ 0.\ \ j{:=}\ \#L.\ \ j{-}h{\ge}1 \Rightarrow Q$$

$$j{-}h{\ge}1 \Rightarrow Q \;\Leftarrow\; \textbf{if}\ j{-}h = 1\ \textbf{then}\ j{-}h{=}1 \Rightarrow Q\ \textbf{else}\ j{-}h{>}1 \Rightarrow Q\ \textbf{fi}$$

$$j{-}h{=}1 \Rightarrow Q \;\Leftarrow\; ok$$

$$\begin{aligned} j{-}h{>}1 \Rightarrow Q \;\Leftarrow\; & j{-}h{>}1 \Rightarrow h'{=}h{<}i'{<}j{=}j'.\\ & \textbf{if}\ abs\ (x - L(i{-}1)) > abs\ (x - L\ i)\ \textbf{then}\ h{:=}\ i\ \textbf{else}\ j{:=}\ i\ \textbf{fi}.\\ & j{-}h{\ge}1 \Rightarrow Q \end{aligned}$$

The comparison $abs\ (x - L(i{-}1)) > abs\ (x - L\ i)$ could just as well have been $abs\ (x - L(i{-}1)) \ge abs\ (x - L\ i)$.

$$j{-}h{>}1 \Rightarrow h'{=}h{<}i'{<}j{=}j' \;\Leftarrow\; i{:=}\ div\ (h{+}j)\ 2$$

For getting the right result, it doesn't matter where i' is, as long as $h{<}i'{<}j$. But for the time bound, it does matter that i' is halfway between h and j .

Proof of the R refinement, starting with the right side.

$$\begin{array}{lll} & h{:=}\ 0.\ \ j{:=}\ \#L.\ \ j{-}h{\ge}1 \Rightarrow Q & \text{replace } Q\\ = & h{:=}\ 0.\ \ j{:=}\ \#L.\ \ j{-}h{\ge}1 \;\Rightarrow\; abs\ (x - L\ h') \;=\; (\Downarrow k\colon h,..j\cdot\ abs\ (x - L\ k)) & \\ & & \text{substitution twice}\\ = & abs\ (x - L\ h') \;=\; (\Downarrow k\colon 0,..\#L\cdot\ abs\ (x - L\ k)) & \text{definition of } R\\ = & R & \end{array}$$

Proof of the $j{-}h{\ge}1 \Rightarrow Q$ refinement, by cases. First case:

$$\begin{array}{lll} & j{-}h{=}1\ \wedge\ (j{-}h{=}1 \Rightarrow Q) & \text{discharge}\\ = & j{-}h{=}1\ \wedge\ Q & \text{specialization}\\ \Rightarrow & Q & \text{generalization}\\ \Rightarrow & \neg\, j{-}h{\ge}1\ \vee\ Q & \text{material implication}\\ = & j{-}h{\ge}1 \Rightarrow Q & \end{array}$$

Last case:

$$\begin{array}{lll} & j{-}h{\ne}1\ \wedge\ (j{-}h{>}1 \Rightarrow Q) \;\Rightarrow\; (j{-}h{\ge}1 \Rightarrow Q) & \text{portation}\\ = & j{-}h{\ne}1\ \wedge\ (j{-}h{>}1 \Rightarrow Q)\ \wedge\ j{-}h{\ge}1 \;\Rightarrow\; Q & \text{combine } j{-}h{\ne}1\ \wedge\ j{-}h{\ge}1\\ = & j{-}h{>}1\ \wedge\ (j{-}h{>}1 \Rightarrow Q) \;\Rightarrow\; Q & \text{discharge}\\ = & j{-}h{>}1\ \wedge\ Q \;\Rightarrow\; Q & \text{specialize}\\ = & \top & \end{array}$$

Proof of $j{-}h{=}1 \Rightarrow Q$ refinement:

$(j–h=1 \Rightarrow Q \Leftarrow ok)$ replace ok and Q

$= \quad (j–h=1 \Rightarrow abs(x – L\,h') = (\Downarrow k: h,..j\cdot\ abs(x – L\,k)) \Leftarrow h'=h \wedge i'=i \wedge j'=j)$

portation

$= \quad h'=h \wedge i'=i \wedge j'=j \wedge j–h=1 \Rightarrow abs\,(x – L\,h') = (\Downarrow k: h,..j\cdot\ abs\,(x – L\,k))$

using context $j–h=1$, $\Downarrow$ has a one-point domain

$= \quad h'=h \wedge i'=i \wedge j'=j \wedge j–h=1 \Rightarrow abs\,(x – L\,h') = abs\,(x – L\,h)$

using context $h'=h$, and reflexive and base laws

$= \quad \top$

Proof of $j–h>1 \Rightarrow Q$ refinement, staring with the right side:

$j–h>1 \Rightarrow h'=h<i'<j=j'$.
if $abs\,(x – L(i–1)) > abs\,(x – L\,i)$ **then** $h:= i$ **else** $j:= i$ **fi**.
$j–h\ge 1 \Rightarrow Q$ distribute $j–h\ge 1 \Rightarrow Q$

$= \quad j–h>1 \Rightarrow h'=h<i'<j=j'$.
if $abs\,(x – L(i–1)) > abs\,(x – L\,i)$ **then** $h:= i.\ \ j–h\ge 1 \Rightarrow Q$ **else** $j:= i.\ \ j–h\ge 1 \Rightarrow Q$ **fi**

replace Q

$= \quad j–h>1 \Rightarrow h'=h<i'<j=j'$.
if $abs\,(x – L(i–1)) > abs\,(x – L\,i)$
then $h:= i.\ \ j–h\ge 1 \Rightarrow abs\,(x – L\,h') = (\Downarrow k: h,..j\cdot\ abs\,(x – L\,k))$
else $j:= i.\ \ j–h\ge 1 \Rightarrow abs\,(x – L\,h') = (\Downarrow k: h,..j\cdot\ abs\,(x – L\,k))$ **fi**

substitution law in **then** and **else** parts

$= \quad j–h>1 \Rightarrow h'=h<i'<j=j'$.
if $abs\,(x – L(i–1)) > abs\,(x – L\,i)$
then $j–i\ge 1 \Rightarrow abs\,(x – L\,h') = (\Downarrow k: i,..j\cdot\ abs\,(x – L\,k))$
else $i–h\ge 1 \Rightarrow abs\,(x – L\,h') = (\Downarrow k: h,..i\cdot\ abs\,(x – L\,k))$ **fi**

I apologize for the informality of this step. Since i is properly between h and j , both $L(i–1)$ and $L\,i$ are in the interval $h,..j$. After the **if** , the comparison says that x is closer in value to $L\,i$ than to $L(i–1)$. Since the list is sorted, that means x is closer in value to some item in $i,..j$ than to any item in $h,..i$. So the closest item in the whole interval $h,..j$ is to be found in the right part $i,..j$. Likewise if the comparison is false, the closest item in the whole interval $h,..j$ is to be found in the left part $h,..i$.

$\Longrightarrow \quad j–h>1 \Rightarrow abs\,(x – L\,h') = (\Downarrow k: h,..j\cdot\ abs\,(x – L\,k))$ definition of Q

$= \quad j–h>1 \Rightarrow Q$

Proof of the $j–h>1 \Rightarrow h'=h<i'<j=j'$ refinement:

$(j–h>1 \Rightarrow h'=h<i'<j=j' \Leftarrow i:= div\,(h+j)\ 2)$ portation

$= \quad j–h>1 \wedge (i:= div\,(h+j)\ 2) \Rightarrow h'=h<i'<j=j'$ replace assignment

$= \quad j–h>1 \wedge h'=h \wedge i' = div\,(h+j)\ 2 \wedge j'=j \Rightarrow h'=h<i'<j=j'$ definition of div

$= \quad \top$

I have not done the timing, but it's a binary search.